The DoD Architecture Framework Views as Requirements Vehicles in a Model Driven Architecture Systems Development Process

Dr. Michael P. Bienvenu, bienvenu@mitre.org, The MITRE Corporation
Keith A. Godwin, kgodwin@mitre.org, The MITRE Corporation

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUN 2004		**00-00-2004 to 00-00-2004**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
The DoD Architecture Framework Views as Requirements Vehicles in a Model Driven Architecture Systems Development Process	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Mitre Corporation,202 Burlington Road,Bedford,MA,01730**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**31**	

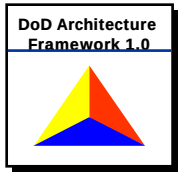
Agenda

- Architecture Framework Overview
- A Systems Engineering Perspective of Architecture Views
- An Integrated Architecture Approach
- Application to JSSEO Model Driven Architecture Development
- Tool Adaptation
- Requirements Management
- Summary

Purpose

- Describe our Approach to Extending DoDAF to Unify Architecture, Requirements and Requirements Traceability
- Demonstrate that the DoDAF can be Inline with the Systems Engineering Process

DoDAF Background



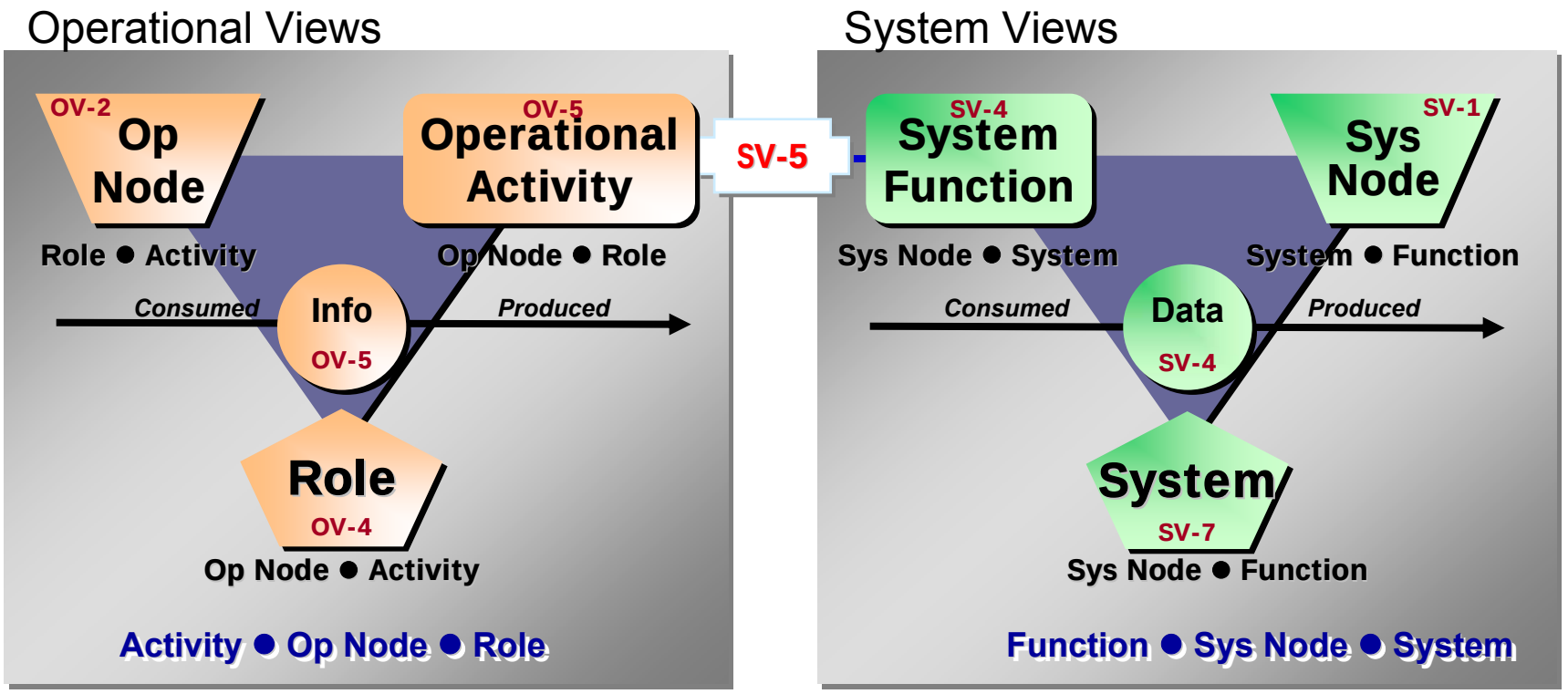
DoD Architecture Framework v1.0

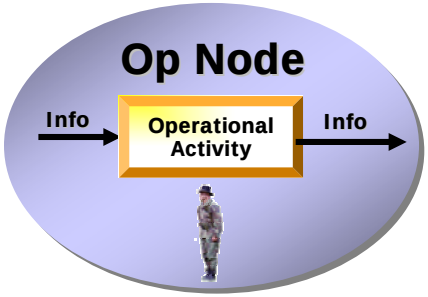
- DoDAF is Mandated for Representing Architectures for the DoD
  - Operational, System, Technical Views (AV, OV, SV, TV)
  - Addresses Structure, Data, Behavior
  - Mainly Diagrams or Tables
- DoDAF is Governed by a Working Group with Representatives from Across DoD Services and Agencies
- *Focus Should Be on the Underlying Meta-Data*
  - What The Diagrams Mean, Not What They Look Like
- Not Intended as a Systems Engineering Tool, or as a Primary Requirements Vehicle
  - Tendency to be Descriptive rather than Prescriptive
  - Doesn't Mandate that Requirements be Specified
  - Assumes (but doesn't require) a Disciplined Process with Strict Consistency Between Products

Challenges for Our Project

- Desire to use DoDAF to Support Systems Engineering
  - Architectures as more than just a Descriptive Report
- Coupled Architectures – Operational, System, Software
- Linked, Traceable Requirements at all Levels
- Address Model Driven Architecture (MDA) Challenges
  - Integrated Architecture Behavior Model (IABM) to meet needs of Single Integrated Air Picture (SIAP)
  - Distributed Nature of the Desired System
  - Rapid Development Prior to Definition of the Full Set of Requirements -- Evolutionary/Iterative Development
  - Iterative Development, Constant Refinement of Requirements
  - OO Based Design Processes Based on UML notation

Relating DoDAF OV and SV Products





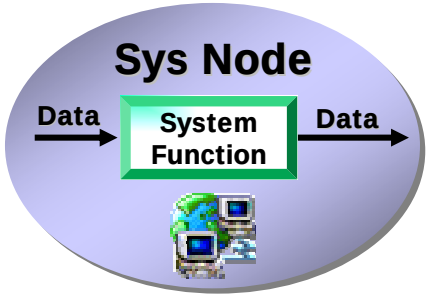
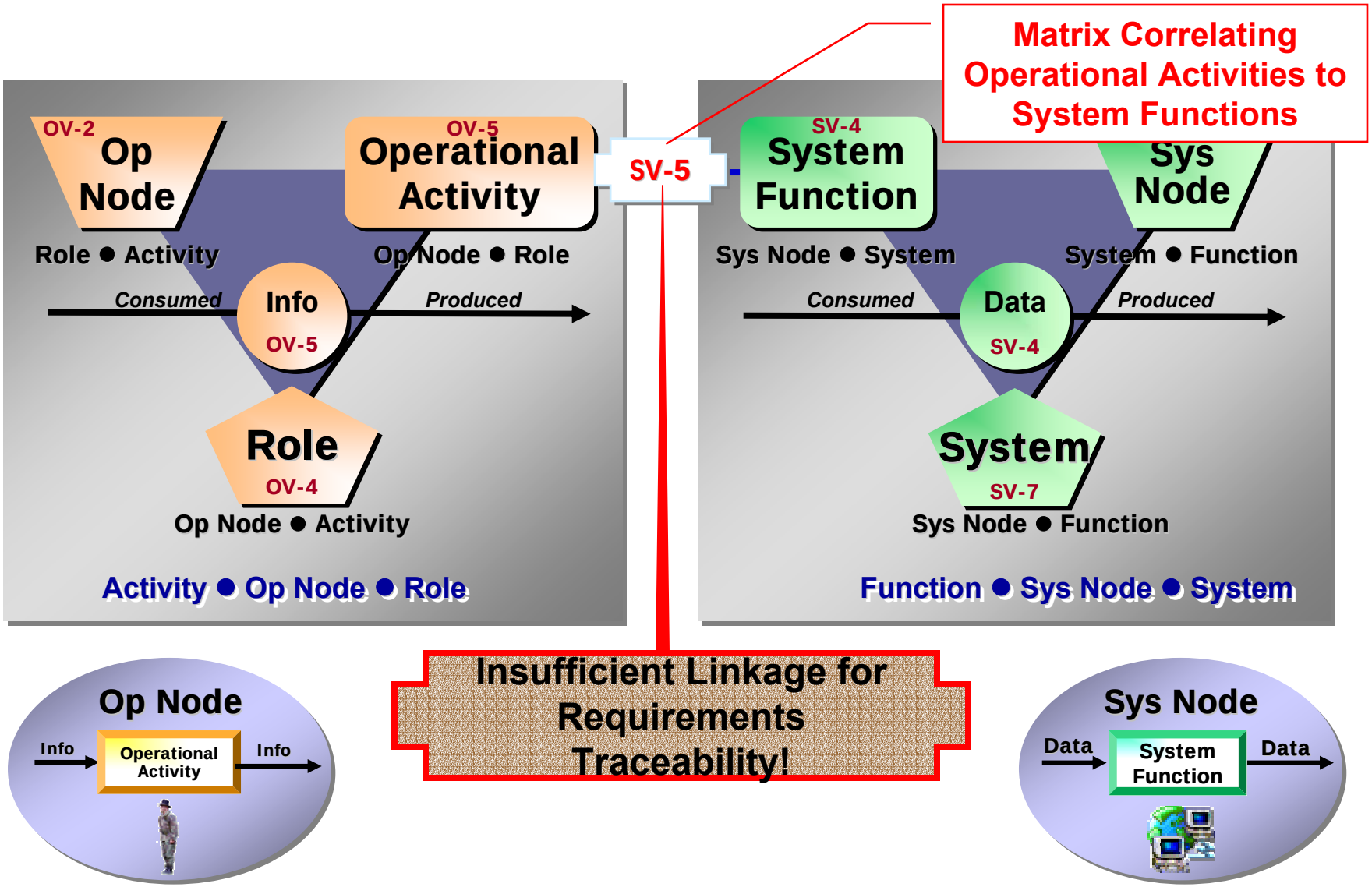


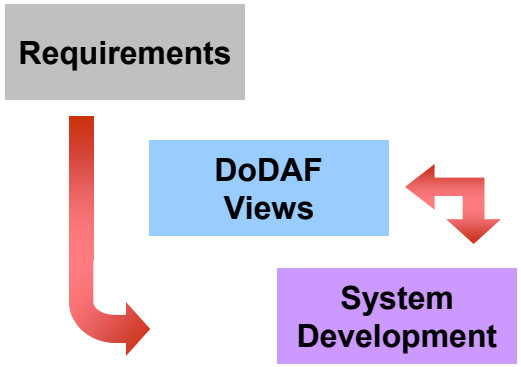
Diagram Concept from Steve Ring, MITRE Corp.

DoDAF OV to SV Connection - *Deficiencies*

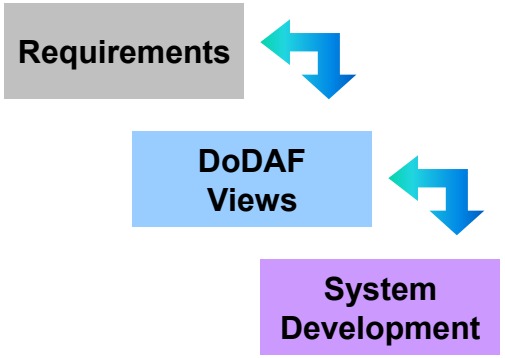


A Systems Engineering View

- Requirements Allocation and Traceability Provide Rigor Needed to have Architecture Views Support System Engineering
- Need to Establish
  - Linkage Between Requirements and Architecture Elements at Each Level
  - Linkage Between Requirements at Different Levels
- Conventional Approach to DoDAF vs Requirements Aligned Approach



Conventional Relationships



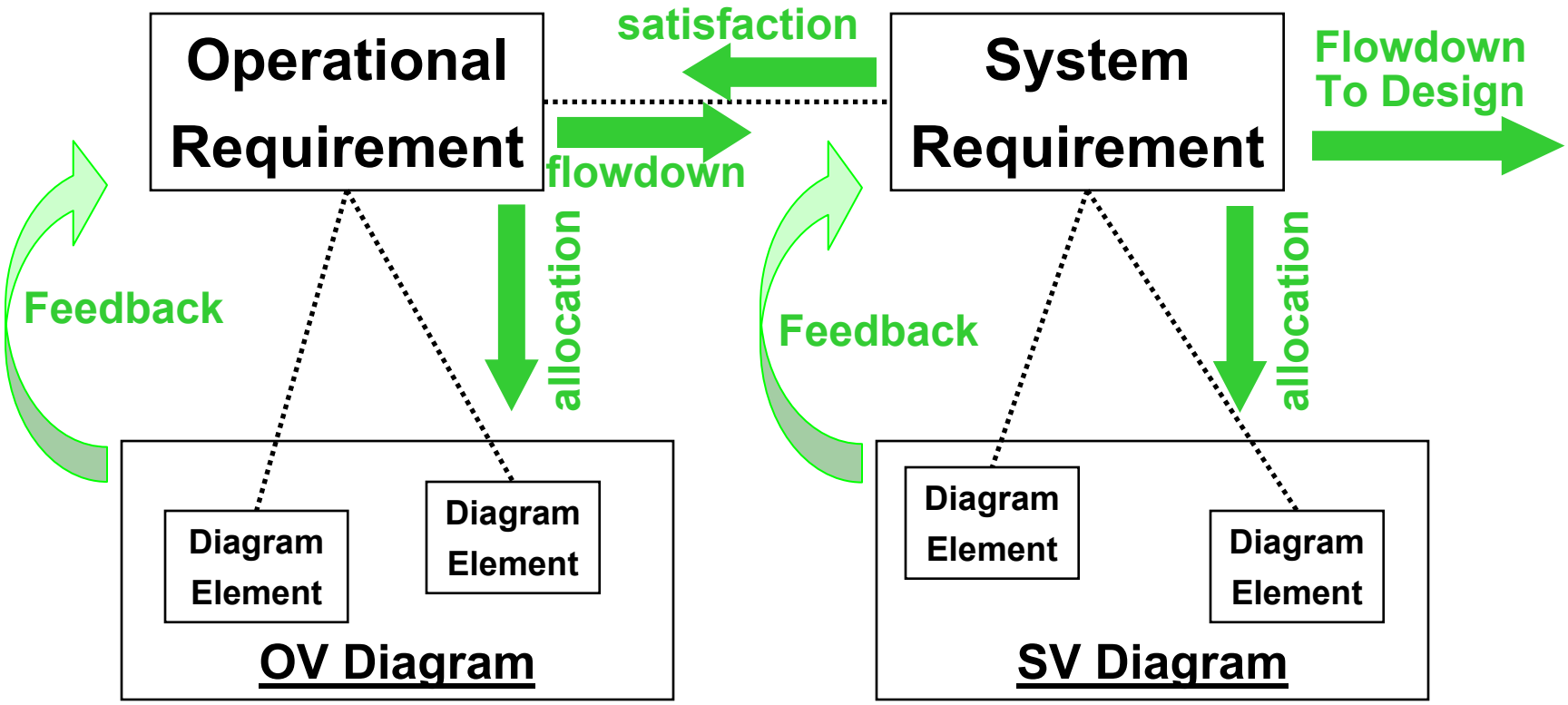
Aligned Requirements Relationships

An Integrated Architecture Approach

- Architectures Capture Requirements in Context
  - Architecture Views are Relevant to the Systems Engineering Process and Become De Facto Living Documents with the Evolutionary System Design
  - Separated Requirements may not have the Meaning they have in Context, or in a Specified Sequence (Using *Rules*, *Statecharts* or *Sequence* Representations)
- All Requirements get Implemented through Something in the Architecture, and there Should be Nothing in the Architecture that isn't there to help Satisfy Requirements
- All Elements in an Architecture Should be Satisfying one or more Requirements
  - Richer and Rigorous Correlation Between Requirements at Different Levels of the Architecture
  - Can be Design-Derived Requirements
- Each Requirement Should be Allocated to at Least One Architecture Element Somewhere
  - If all Requirements Should be Testable, then there must be Something to Test

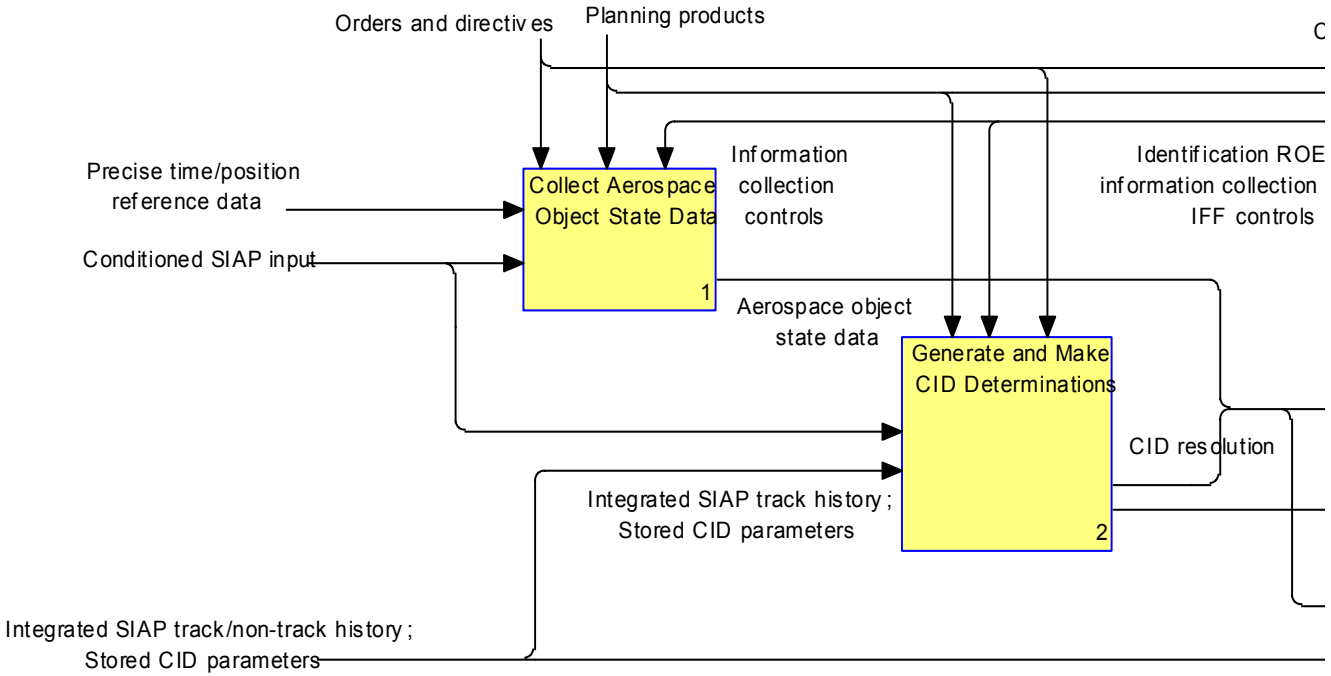
Architecture – Requirements Traceability

- Requirements Apply to More than Just Functions
- Data, Interfaces, and Behavior Should also have Requirements, and be Related between OVs and SVs



Assigning Requirements to an Operational Activity (Example)

- Portion of one of the OV-5 Dataflow Diagrams
- Requirements can be Attached to
  - Operational Activities (boxes)
  - Information Exchanges {data} (lines)



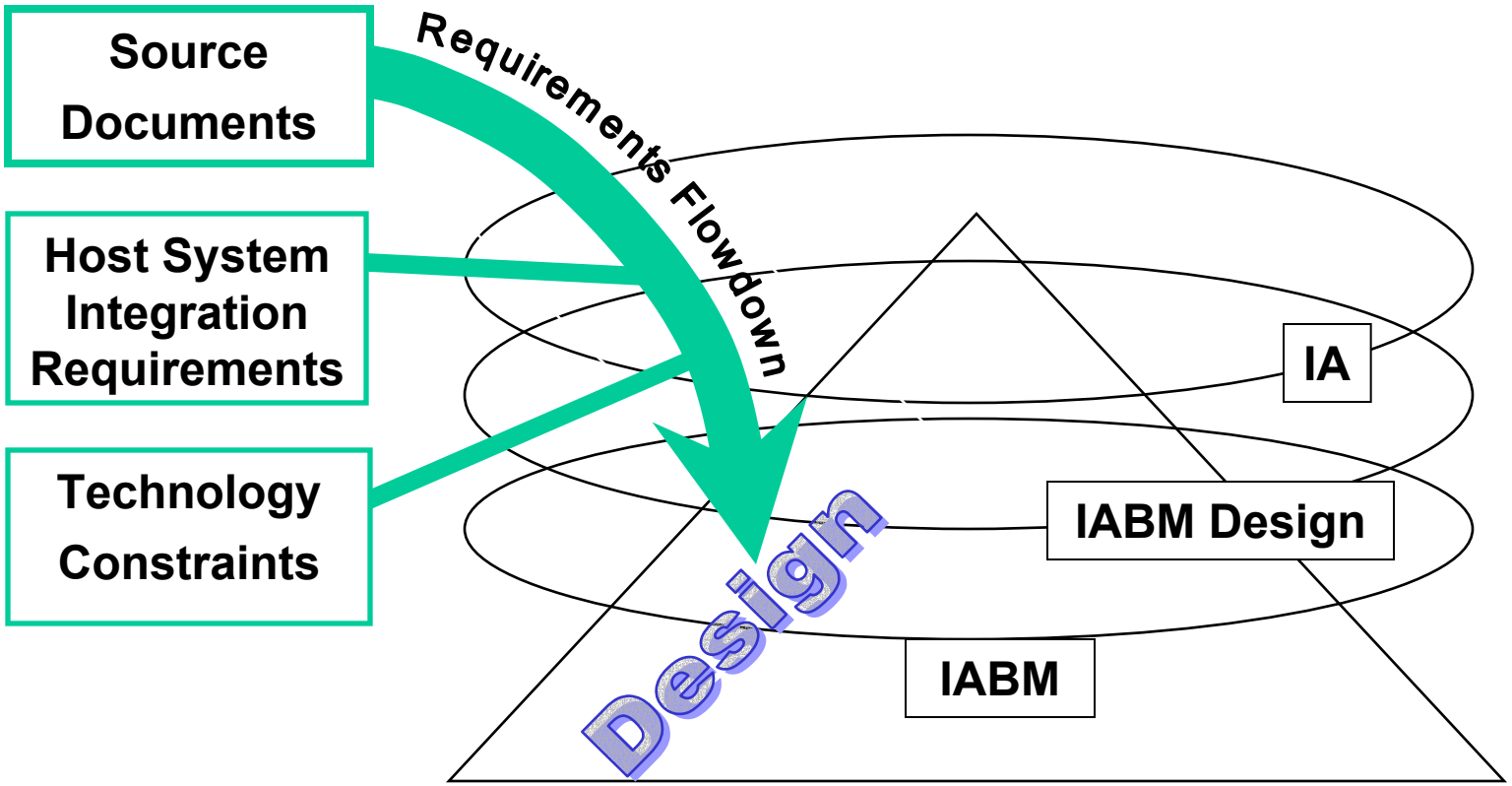
Comprehensive Approach for JSSEO

- JSSEO Project Characteristics
  - Based on Model Driven Architecture (OMG)
  - One Fact, One Place
  - Requirements Traceability
  - Auto-Generation of Documentation
- Agile Development
  - Iterative Requirements Definition and Refinement
  - Appropriate for "Disruptive Systems" *
  - Distributed System and System Requirements
- Support Implementation of Software to Heterogeneous Host Systems
- Tailoring of DoDAF products
  - UML as Basis for System Views

* Clayton M. Christensen, *The Innovator's Dilemma*

Role of an Integrated Architecture within JSSEO

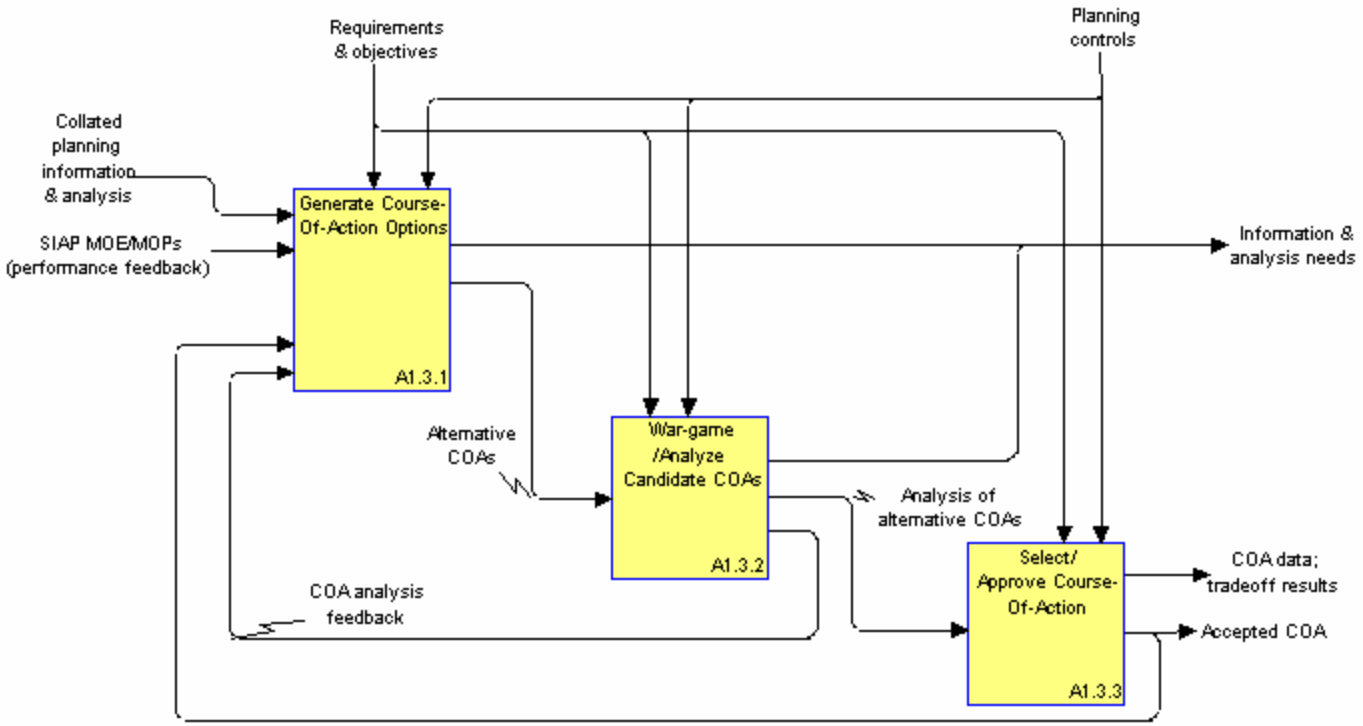
- Integrated Architecture (IA) Contains
  - Operational and System Architecture
  - Operational and System Requirements



Extending DoDAF to Address JSSEO MDA

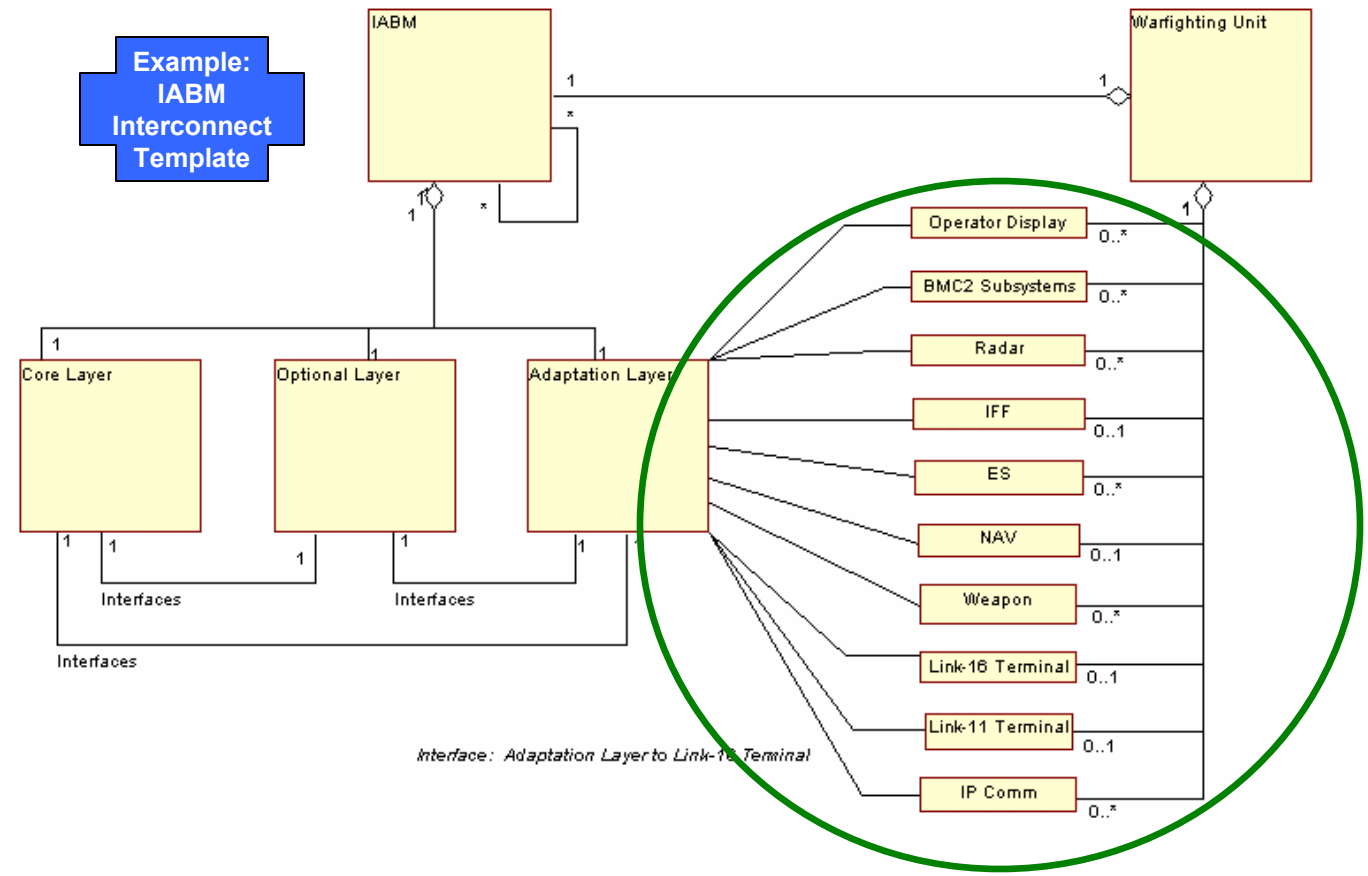
- Diagram Adaptation Primarily on the SV side.
- SV-1, -2, -4, -6, -11: Use UML Class and Object Diagrams
  - Variety of Uses
- Interconnect Template
  - The IABM, its Layers, and its Interfaces to the Host System
  - Classes Defined for Commonality
  - Object Instance Versions for each Host System
- Capability Areas
  - “Virtual” Classes Defined to hold Domain-Level Requirements
- Interface Specification
  - Associations/Links can have Requirements Attached, and Support Message Definition

IA Operational Views & Requirements



- Operational Requirements (Derived from Primary Sources) Associated with Diagram Elements

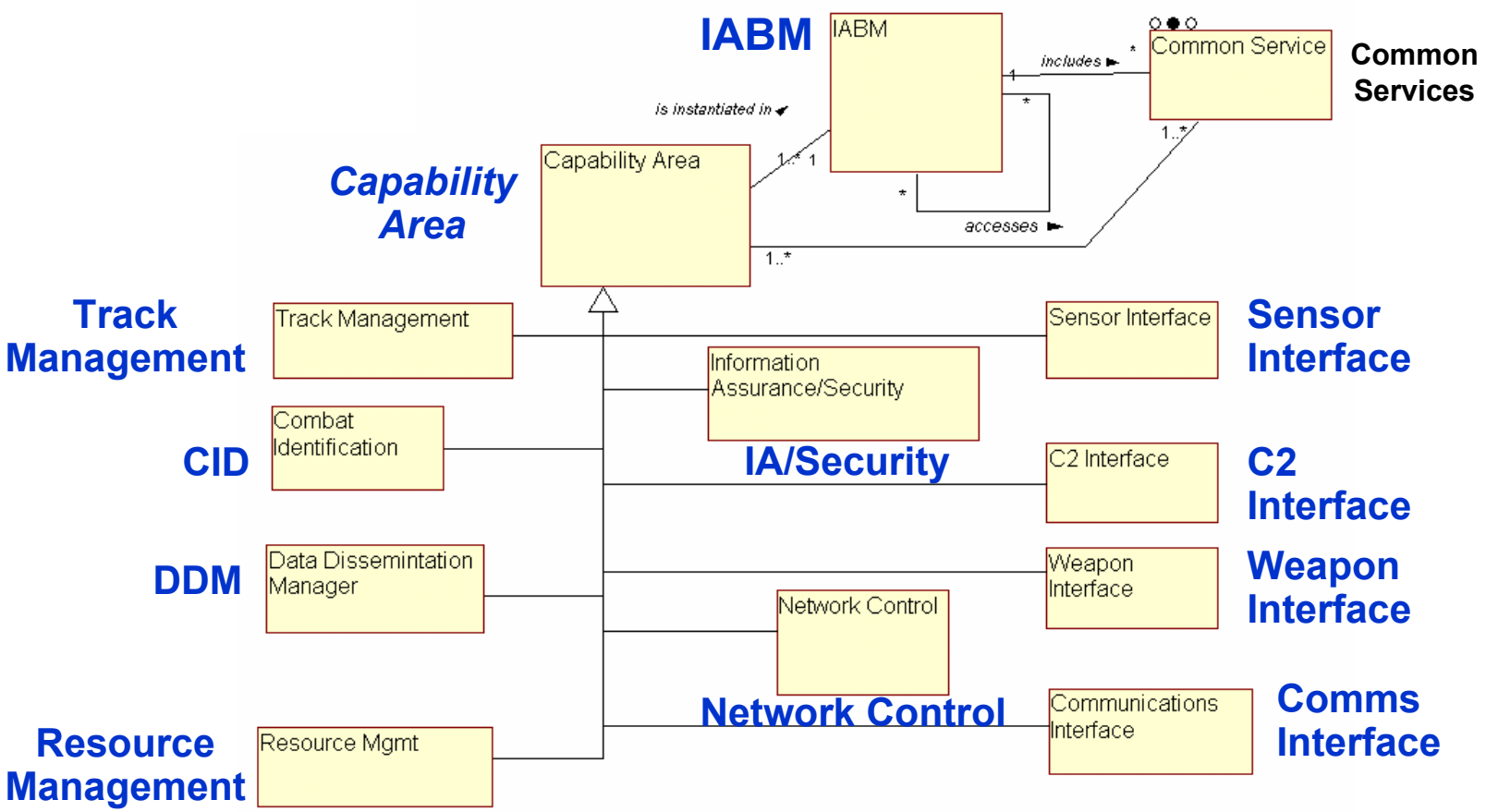
IA System Views & Requirements



- System Requirements Allocated to System View Elements
- Some Requirements derived from Architecture Context (Interfaces)

IABM Capabilities Object Model (Example)

- Links System Views with IABM Design
- IABM "Capabilities" are Virtual Objects used to hold Sets of Related Domain* Requirements.



* Kennedy-Carter iUML Development Tool Terminology for a Collection of Classes

Value of Architecture/Requirements Process to JSSEO

- Unified Repository of Integrated Information
  - Allows Automated Detection of Mismatches
  - Support for Automated Document Generation
  - Integrated Product Focus for Configuration Control & Management
- Efficiency: Engineers Think, Tools Help Keep Track
- Fewer Tools Means Fewer Manual Translations between Tools
  - Every (manual) Translation Provides an Opportunity for Mis-translation
  - Translations Mean More Effort, More Complicated Updating Process, Lower Probability of Continued Success
- Up-To-Date Design
  - Architecture, Requirements, Design Updated Monthly

Implementing the Solution

- Architecture Tool Adaptation
  - Architecture Diagramming and Requirements Management Tools Configured to Support the JSSEO Development Process
  - Automated Data Exchange Between Tools to Minimize Data Entry Duplication
- Requirements Management
  - Flexible Scheme for Identifying and Tracing Requirements
  - Requirements Managed Individually, not as a Set within a Specification
- Metrics, Reports and Status Monitoring
  - Oriented Toward Determining Completeness of Requirements Traceability
  - Account for All Aspects of Traceability
    - Requirement to Source
    - Requirement to Requirement
    - Requirement to Architecture View Diagram Elements
    - Requirement to Development Tool Domains

Adapting Tools

- No Single Tool Meets All Needs - Requires Suite of Interoperable Tools

PRIMARY TOOLS

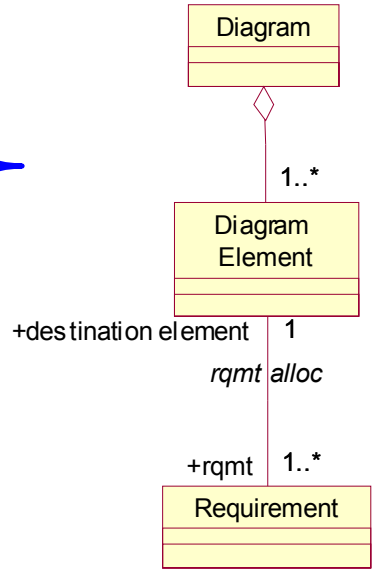
- Popkin Systems Architect
  - DoDAF Views (Diagrams)
  - Requirements (multiple levels)
  - Associates Requirements with Architecture Elements (Symbols & Definitions)
  - Encyclopedia of Architecture Data Stored in MS SQLServer
- Telelogic DOORS
  - Requirements Repository
  - Traceability Management
  - Interface to Pass Requirements into Kennedy-Carter iUML Development Tool

SUPPORTING TOOLS

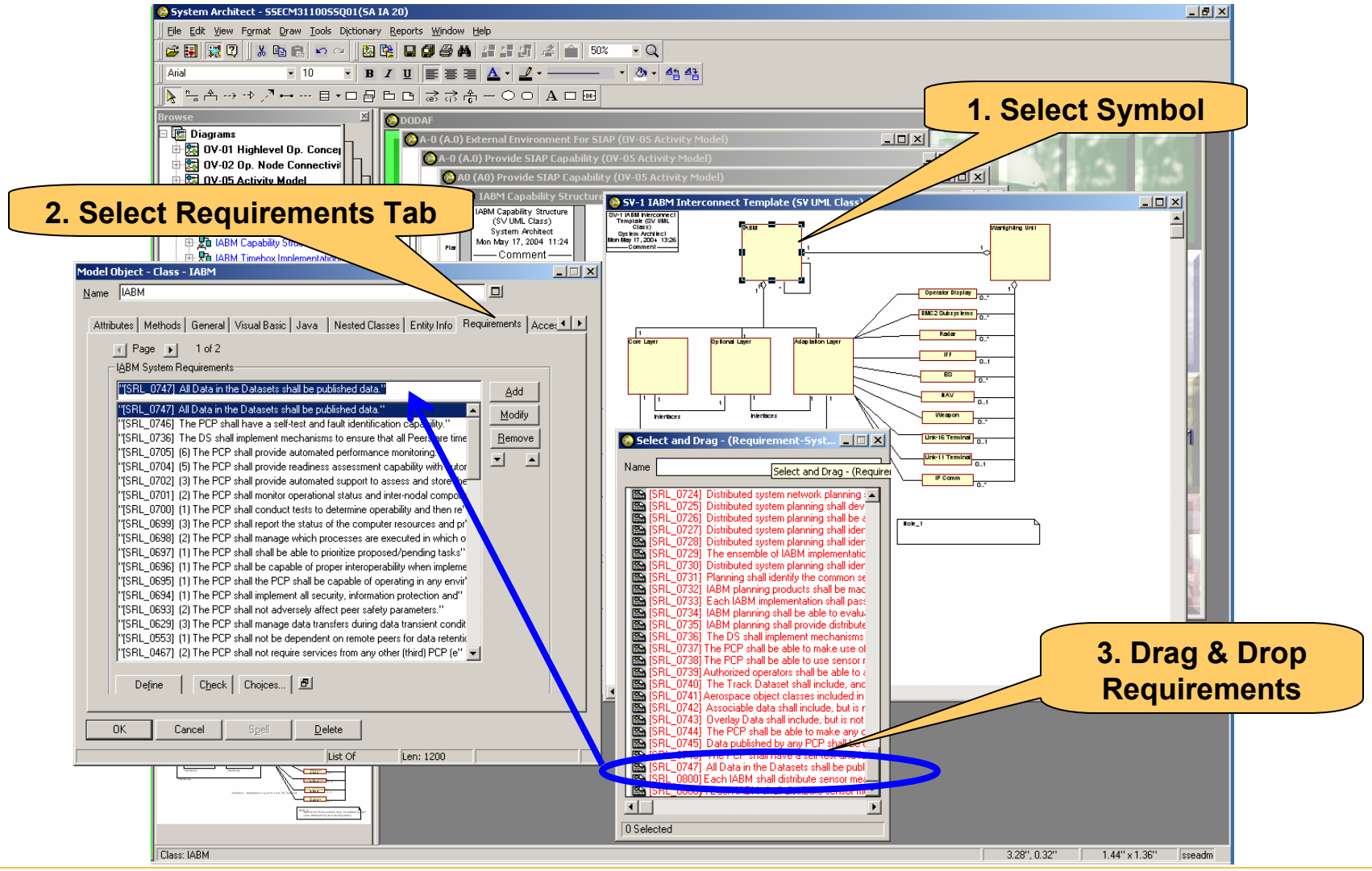
- MS Excel
  - CSV Files for Export/Import of Requirements Between DOORS & System Architect
- MS Access
  - Statistical Reports on Requirements Management
  - SQLServer Import/Export of Architecture Data
- HTML
  - Browser Viewable Reports of Architecture Elements and Associated Requirements

Popkin System Architect

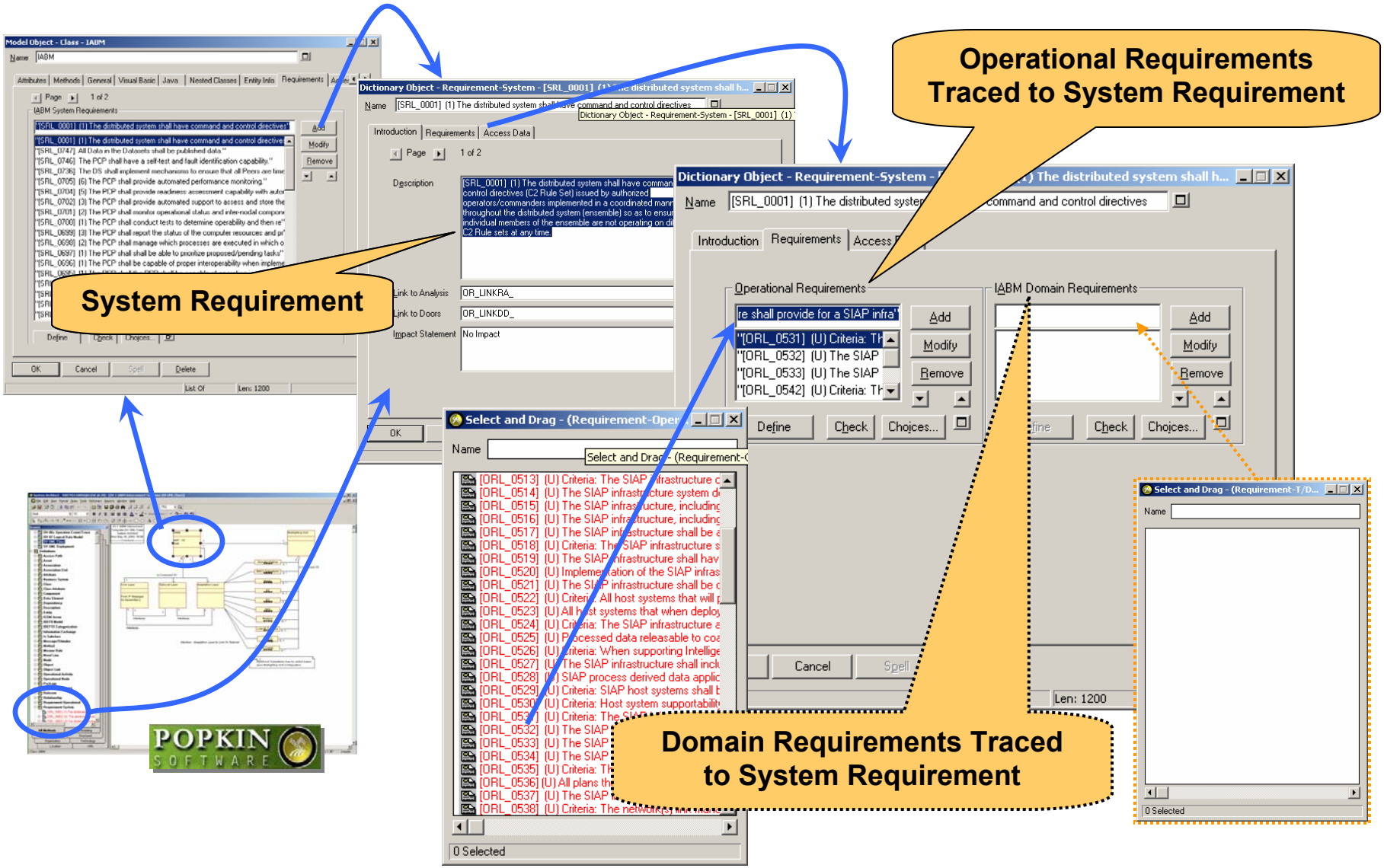
- Configured for JSSEO Development Process
  - UML for System Views to Align with UML in MDA
- Modified USRPROPS.TXT file
  - Added Requirement Definitions (Addressables) for Operational, System, and Domain Requirements
  - Extended Symbol Definitions to Accept Associations of Requirement Addressables
  - Extended System Requirement Definition to Accept Associations of Operational and Domain Requirement Addressables
- Used to Build DoDAF OVs and SVs
- Imports Requirements from DOORS Repository (via Excel Files)
- Assigns Requirements to Diagram Elements
  - Drag and Drop Requirements to Diagram Symbols



Attaching Requirements To A Symbol



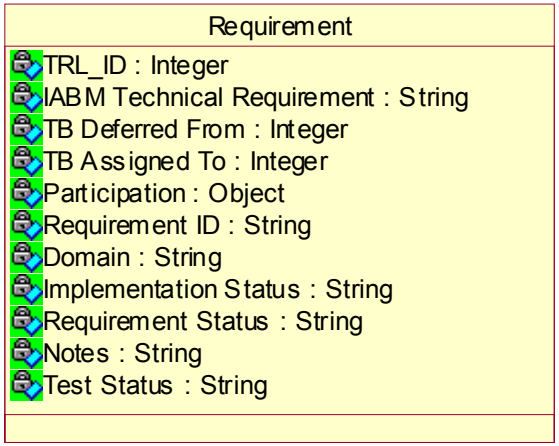
Defining Requirements Linkages



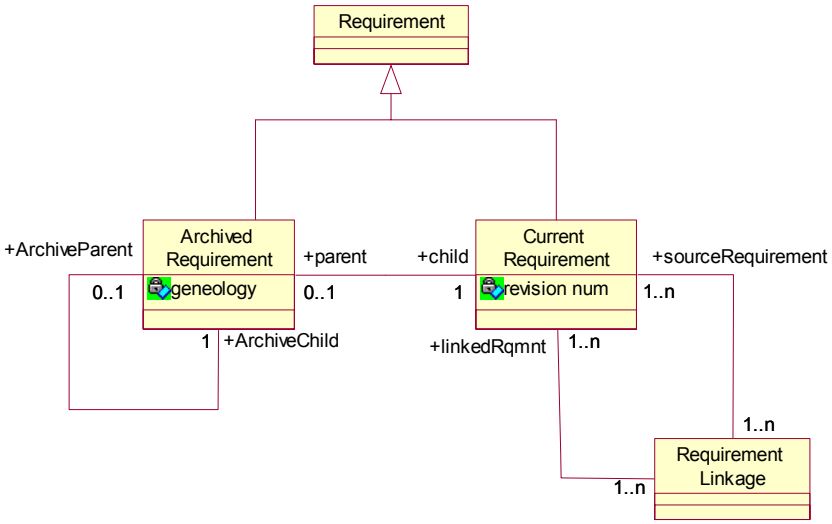
MITRE

Requirements Management

- Requirements Database in DOORS
  - Independent Operational, System, and Domain Requirement Lists
  - Unique Identifier for Each Requirement
  - Requirement Attributes for Status Tracking
- Traceability to Source Documents, Between Requirements and to Architecture Elements
- Reports on “Orphan” Requirements or Architecture Elements Produced from both SA and DOORS

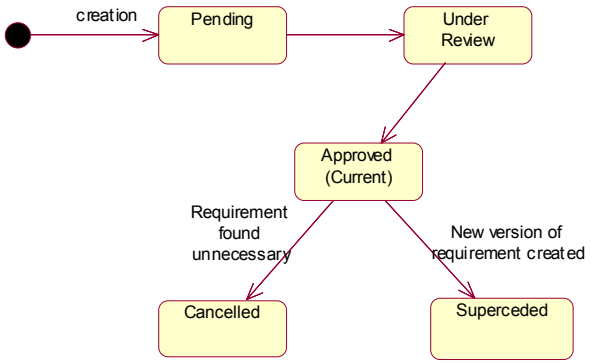


Requirement Internal Meta Structure & States



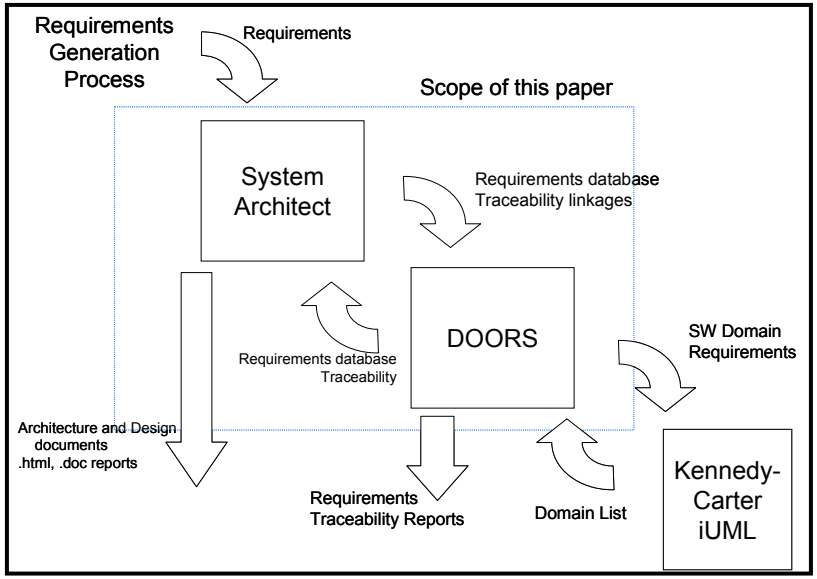
- Only Current Requirements are Linked
  - Linkages to Diagram Element, Other Requirements, or Source Document
- Superseded or Cancelled Requirements are Archived

Requirements Class State Chart



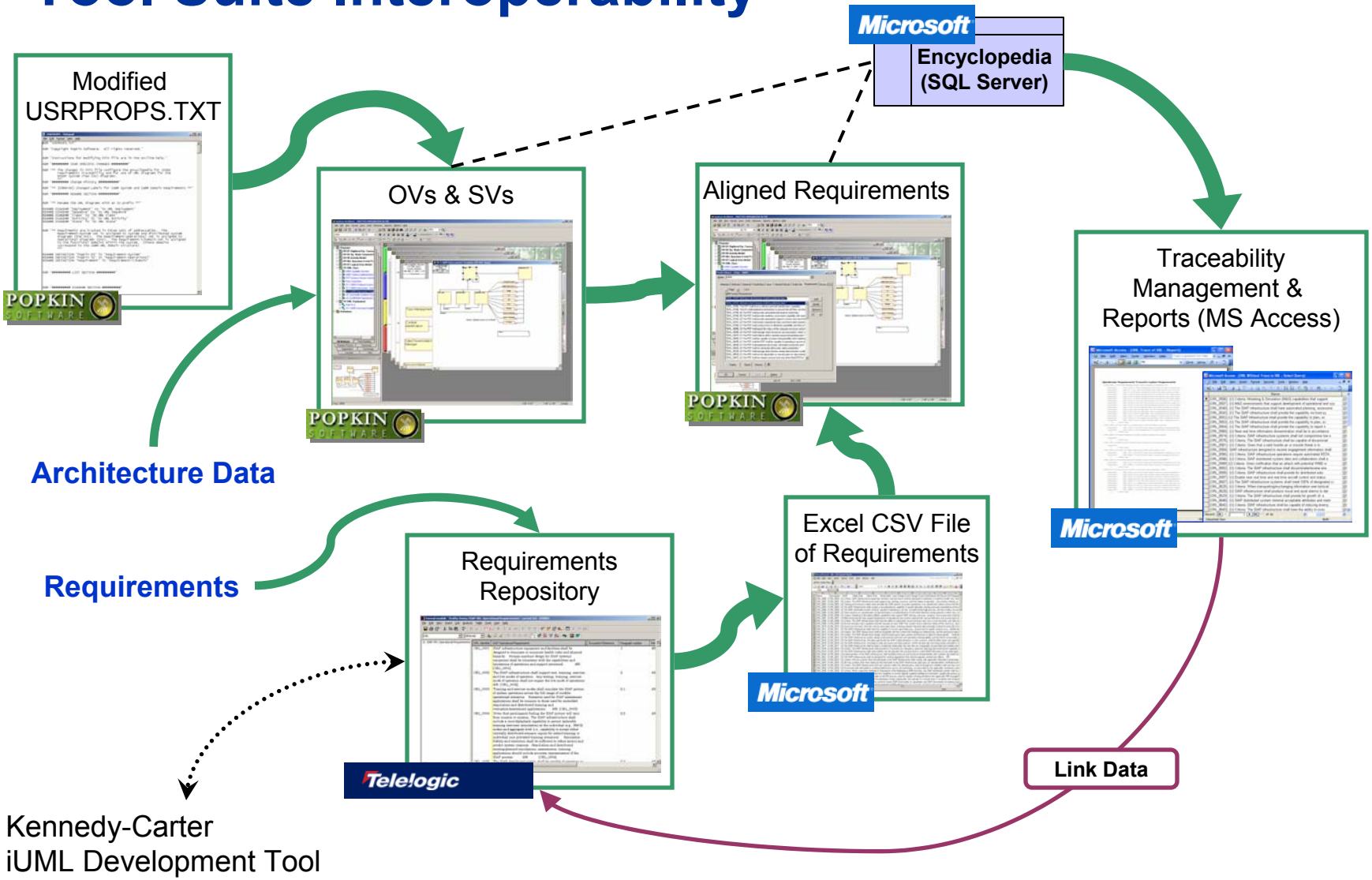
- Requirements are Approved Prior to Assigning Linkages
- Requirements, Once Created, Stay in System

Requirements Work Flow



- **Movement of Requirements Between Tools Requires Adaptation of 'One Fact One Place' Program Goal**
  - **Requirement Definition in DOORS**
    - **Exported to System Architect**
  - **Requirement Relationships Defined in System Architect**
    - **Exported for Detailed Reporting**
    - **Exported to DOORS for Traceability Management**

Tool Suite Interoperability



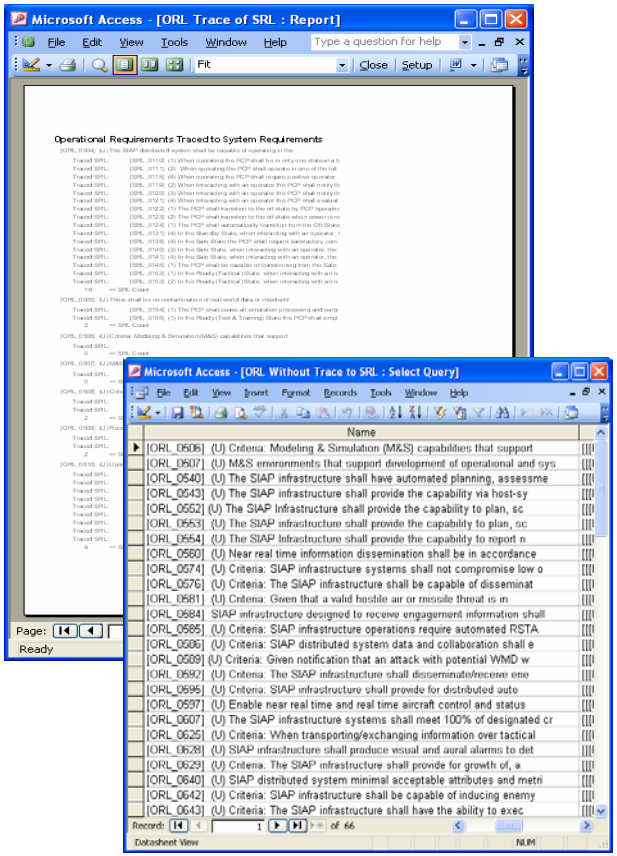
Metrics and Reports

- Measuring the Goodness of Traceability
  - Completeness of Architecture Views
  - Completeness of Requirements Set
- Traceability Statistical Reports
  - Used to Assess Architecture and Requirements Traceability
  - Requirements Traced into the Architecture
  - Architecture Elements Aligned with a Requirement
- HTML Reports from System Architect and DOORS
  - Provides Access to Architecture and Requirements Information without Requiring Expertise in Tools
  - Permits Wide Review Without need for Special Tools

Reports used to Improve Overall SIAP Development Process

Requirements Traceability Reports

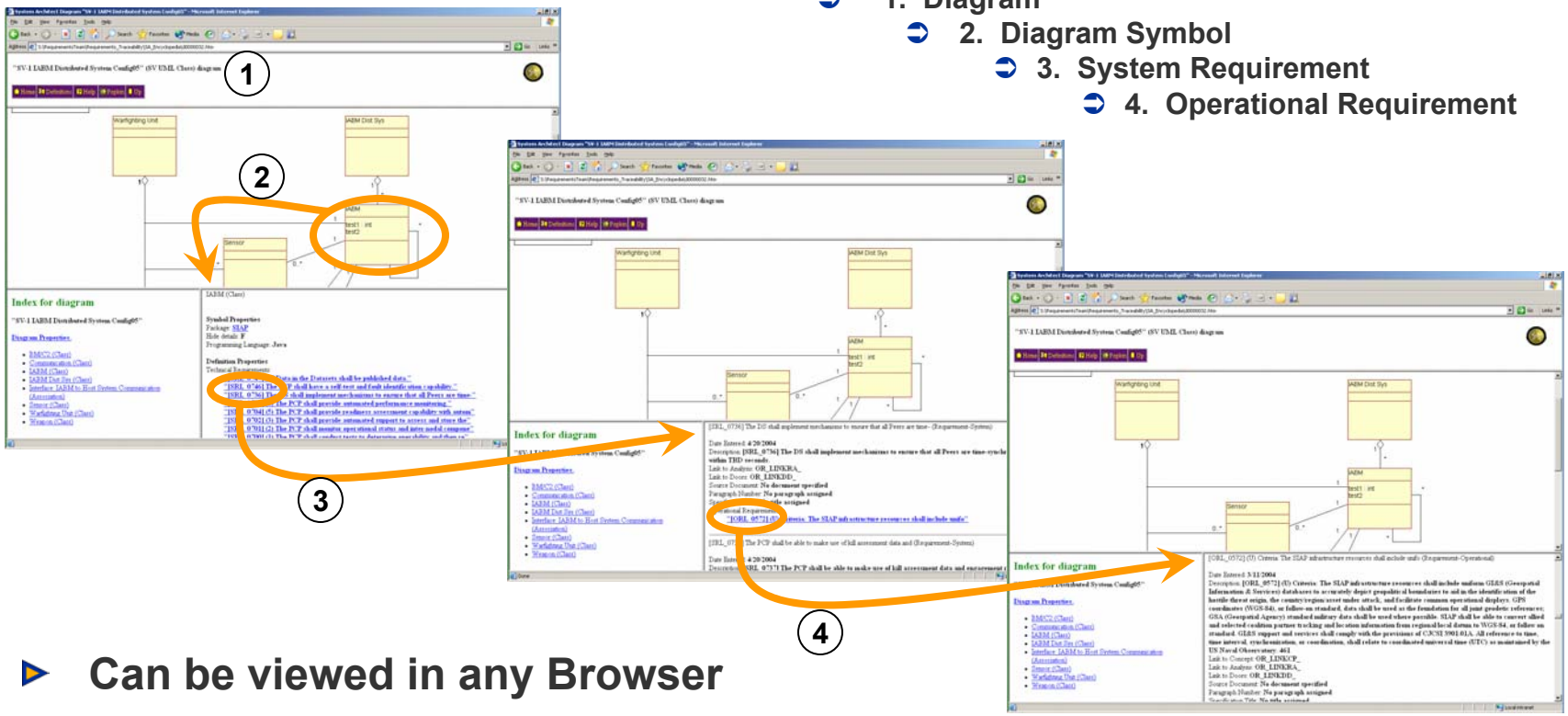
- Reports Built in MS Access using Data Extracted from System Architect Encyclopedia (MS SQL Server)
  - Architecture and Requirements Traceability
  - Requirements Accounted for in the Architecture
  - Architecture Elements with Assigned Requirement



MITRE

HTML Reports

- 1. Diagram
  - 2. Diagram Symbol
    - 3. System Requirement
      - 4. Operational Requirement



- Can be viewed in any Browser
- Hyper-Linked Data
  - Symbols to Symbol Definition (Includes Assigned Requirements)
  - Requirement Name to Corresponding Definition
  - Requirements to Requirements

Future Work

- Comprehensive Hyper-linking
  - VB Scripts used to create hyper-linked Integrated Architecture
  - Linkages with HTML from DOORS and iUML Tools
  - Complete Requirements Trace From Source Documents to IABM Domain Classes
- Additional Reporting & Analysis Features
- Direct Database Exchanges to Minimize need for File Export/Import to Move Data between Tools

Summary

- Presented the Approach for Linking Architecture and Requirements.
  - Architecture Views Serve to Place Requirements in Context
- Demonstrated the Current State of **JSSEO** Products, Metrics, Reports
- Metadata Structure, Configuration Data in the usrprops.txt file, is U.S. Government Owned, and Releasable (through JSSEO)
- Requirements Must Be Developed By All, Integrated With Architecture

Requirements, Design, or Behavior that is not Part of an Integrated Architecture is not Defensible